AIROC™ CYW9x907 Wi-Fi 接続プロセッサの低電力モード

本書について

適用範囲と目的

このアプリケーションノートでは、AIROC™ CYW9x907 Wi-Fi 接続プロセッサの低電力モードを紹介し、低電力モードを使用するためにアプリケーションのビルドを調整する方法について説明します。

対象者

このアプリケーションノートは、次のいずれかのキットで、低電力サポートを使用するアプリケーションを実行するユーザーを対象とします。

1. CYW943907AEVAL1F
2. CYW954907AEVAL1F

目次

本書について **1**
目次 **2**
1 はじめに **3**
2 低電力モード機能 **4**
2.1 アプリケーションプロセッサ (APP CPU) 4
2.2 ワイヤレス LAN プロセッサ (WLAN CPU) 4
3 APP CPU 低電力モード設定 **5**
3.1 スリープ 5
3.2 ディープ スリープ 5
3.2.1 ディープスリープのレイテンシ 5
3.3 ハイバネーション 6
4 WLAN CPU 低電力モード **7**
4.1 デフォルトの WLAN 低電力モード 7
4.2 PM1 省電力モード 7
4.3 PM2 省電力モード 7
5 HAL 関数 **8**
5.1 シングルスレッドアプリケーション 8
5.1.1 SW1 割込みを使用するシングルスレッドアプリケーション 8
5.1.2 ティックレススリープまたはディープスリープを使用するシングルスレッドアプリケーション 9
5.2 マルチスレッドアプリケーション 10
5.3 ディープスリープのロック/ロック解除 10
参考資料 **11**
改訂履歴 **12**
免責事項 **13**

1 はじめに

AIROC™ CYW9x907 Wi-Fi 接続プロセッサは、IEEE 802.11 b/g/n 仕様で規定されているすべてのレートをサポートし、Arm® Cortex®ベースのアプリケーションプロセッサ, シングルストリーム IEEE 802.11n MAC/ベースバンド/無線, デュアルバンド (2.4 GHz/5 GHz) 送信パワーアンプ (PA) , および受信低ノイズアンプ (LNA) を備えています。Figure 1 に、CYW43907 内のすべての主要な物理ブロックとそれに関連する外部インターフェースの相互接続を示します。

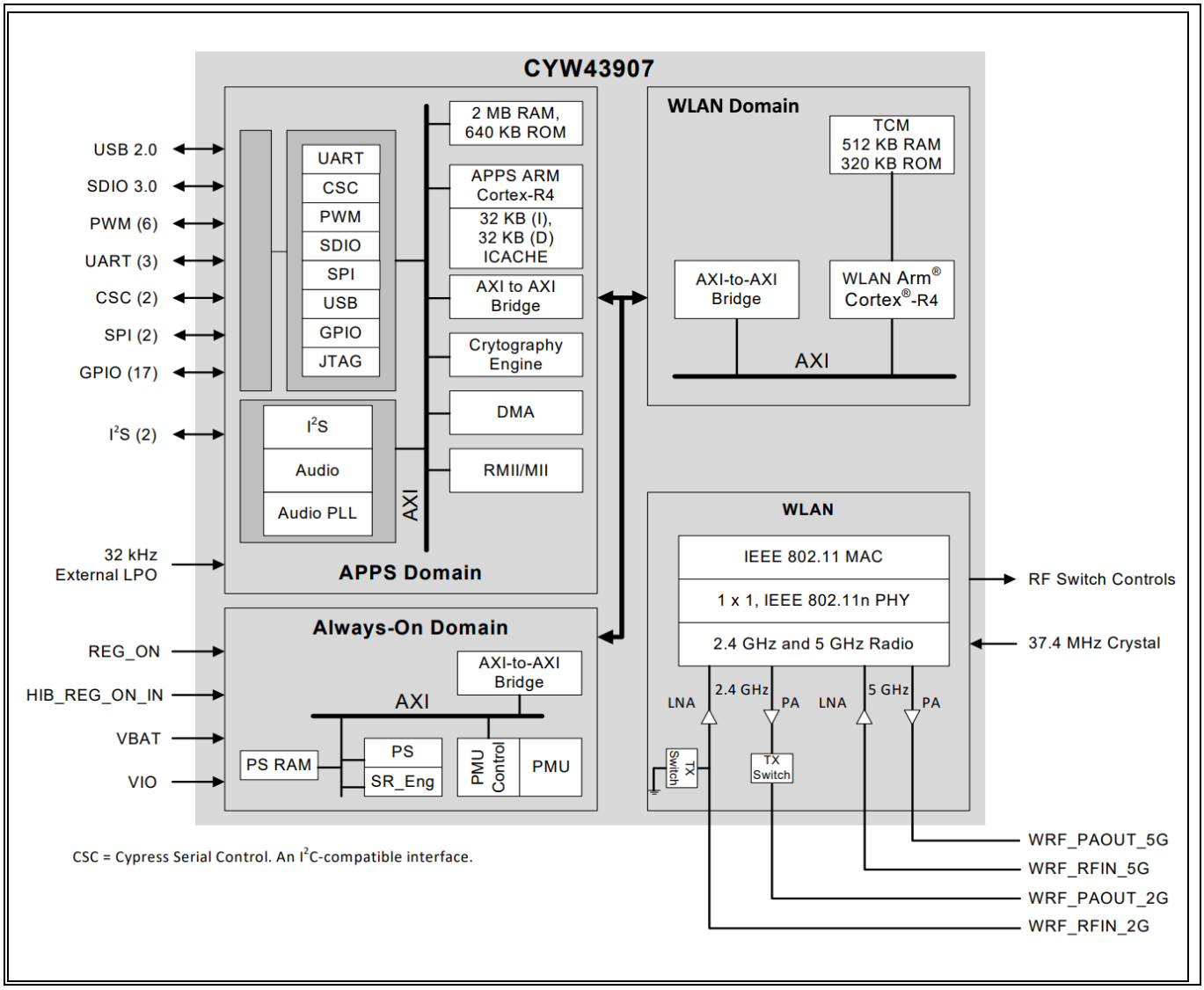


Figure 1 CYW43907 の機能ブロック図

2 低電力モード機能

この MCU プラットフォームには、次の 2 つの CPU が搭載されています。

1. アプリケーションプロセッサ (APP CPU)
2. ワイヤレス LAN プロセッサ (WLAN CPU)

両方の CPU の電源モードは独立しており、一緒に使用することも、別々に使用することもできます。

2.1 アプリケーションプロセッサ (APP CPU)

APP CPU は、ユーザーアプリケーションを実行します。2 つのスリープモード (スリープとディープスリープ) は、プロセッサを短時間停止させることで電力を削減するために使用されます。これは、RTOS と組み合わせて自動的に行われます。さらに、CPI にはハイバネーション低電力モードがあり、指定された時間プロセッサを停止し、ハイバネーション時間が経過するとプロセッサを再起動します。

2.2 ワイヤレス LAN プロセッサ (WLAN CPU)

WLAN CPU は、独立した CPU で、M2M DMA インターフェースを介して APP CPU と通信し、Wi-Fi 無線を制御します。APP CPU は、 WLAN CPU を直接制御しません。コマンドとデータの送受信のみを行います。WLAN CPU は、独自の内部低電力手法を備えています。アプリケーションは、APP CPU から、WLAN CPU が低電力モードに入ることを選択した場合に、そのモードに入ることができることを示すコマンドを送信できます。さらに、アプリケーションは、WLAN CPU が使用する低電力モードを選択する要求を送信できます。

3 APP CPU 低電力モード設定

APP CPU 低電力モード設定は、*<app_root_directory>/bsps/TARGET_APP_CYW943907AEVAL1F/cybsp.h* にあります。

Table 1 APP CPU 低電力設定

定義	注意事項
CY_CFG_PWR_MODE_ACTIVE	アクティブモード – スリープモードに移行しません
CY_CFG_PWR_MODE_SLEEP	スリープモード – プロセッサをスリープ状態にすることで電力を削減します
CY_CFG_PWR_MODE_DEEPSLEEP	ディープスリープ – スリープモード+システムクロックを停止します
CY_CFG_PWR_DEEPSLEEP_LATENCY	ディープスリープに入る前に必要なティック数

Code Listing 1 cybsp.h での電力モード設定

```
#define CY_CFG_PWR_MODE_ACTIVE          (0x00UL)
#define CY_CFG_PWR_MODE_SLEEP           (0x01UL)
#define CY_CFG_PWR_MODE_DEEPSLEEP       (0x02UL)

#ifndef CY_CFG_PWR_SYS_IDLE_MODE
#define CY_CFG_PWR_SYS_IDLE_MODE        (CY_CFG_PWR_MODE_ACTIVE)
#endif

#define CY_CFG_PWR_DEEPSLEEP_LATENCY    (24UL)  /* Min. 24 for this device */
```

3.1 スリープ

スリープは、プロセッサを停止し、システムをフルパワーに戻す割込みが発生するまで、消費電力を約5%削減します。有効にすると、RTOS は、ユーザーアプリケーションがアイドル状態になるタイミングを判断し、APP CPU をスリープモードにして、全体的な消費電力を削減します。

3.2 ディープ スリープ

ディープスリープは、システムクロックを停止し、システムをフルパワーに戻す割込みが発生するまで、消費電力を 75%削減します。有効にすると、RTOS は、アプリケーションがアイドル状態になるタイミングを判断し、APP CPU をディープスリープモードにして、全体的な消費電力を削減します。

3.2.1 ディープスリープのレイテンシ

レイテンシは、ディープスリープに入るのに十分な時間があるかどうかを判断するために使用され、ミリ秒単位で測定されます。これは、RTOS を使用するマルチスレッドアプリケーションでのみ使用されます。RTOS は、次のタスクが実行されるまでの時間を決定します。その時間がレイテンシの値を超えると、システムはディープスリープモードになります。

3.3 ハイバネーション

ハイバネーションは、外部ソースを使用してシステムをウェイクします。ハイバネーション状態になると、デバイスはすべての周辺機器をシャットダウンし、デバイスがウェイクアップするタイミングに関するある程度の制御を維持しながら、消費電力を可能な限り低い値に減らします。これは、デバイス (RAM など) の現在の状態が失われることを意味します。Table 2 に、ハイバネーションのウェイクアップソースを示します。ハイバネーション状態モードに入るために必要なコードを Code Listing 2 に示します。

Table 2 ハイバネーションのウェイクアップソース

定義	注意事項
CYHAL_SYSPM_HIBERNATE_WDT	WDT 割込みをウェイクアップソースとして設定します。 これは、現在サポートされている唯一のウェイクアップソースです。

Code Listing 2 main() からハイバネーションの呼び出し

```
#include "cyhal_syspm.h"

cy_rslt_t result;
result = cyhal_syspm_hibernate(cyhal_syspm_hibernate_source_t wakeup_source);
```

4 WLAN CPU 低電力モード

WLAN CPU は、デバイスの電源投入時にロードされる固定ファームウェアイメージを実行します。WLAN CPU は、APP CPU とは独立して動作し、専用の M2M DMA チャネルを介して APP CPU と通信します。APP CPU で実行されているアプリケーションは、WLAN CPU が低電力モードに入るタイミングを決定できるようなコマンドを送信できます。

Table 3 WLAN CPU 電力モード設定

定義	説明
CY_WCM_NO_POWERSAVE_MODE	省電力モードではありません。
CY_WCM_PM1_POWERSAVE_MODE	スループットの低下を考慮しない指定インターフェースの省電力モード。
CY_WCM_PM2_POWERSAVE_MODE	高スループットを考慮した指定インターフェースの省電力モード。

4.1 デフォルトの WLAN 低電力モード

WLAN 低電力モードのデフォルトは、CY_WCM_NO_POWERSAVE_MODE です。モードを変更するためには、アプリケーションは、WLAN が電源モードを変更することを許可するように関数を呼び出す必要があります。

Code Listing 3 ユーザーアプリケーションでの WLAN 電力モードの許可

```
#include "cy_wcm.h"

Cy_rslt_t result;
result = cy_wcm_allow_low_power_mode(CY_WCM_PM1_POWERSAVE_MODE);
```

4.2 PM1 省電力モード

このモードでは、データスループットが多少犠牲になりますが、電力が最も節約されます。Wi-Fi 接続の整合性は損なわれませんが、一部の操作では「省電力なし」モードよりも若干時間がかかる場合があります。

4.3 PM2 省電力モード

このモードは、PM1 よりも電力の節約が少ないですが、データスループットをより高いレベルに保ちます。Wi-Fi 接続の整合性が損なわれることはありません。

5 HAL 関数

ハードウェア アブストラクション レイヤー (HAL) API は、アプリケーションにデバイスの電力モードに作用する関数を提供します。詳細については、ハードウェア アブストラクション レイヤーの電源管理 API を参照してください。

スリープまたはディープスリープが発生すると、コードは Arm® Cortex®-R4 のソフトウェア割込み命令 (SWI) を実行します。この命令は、割込みによって続行が許可されるまでプロセッサを停止します。

5.1 シングルスレッドアプリケーション

シングルスレッドアプリケーションは RTOS を使用せず、スリープ関数を直接呼び出す必要があります。アプリケーションは、APP CPU を再開する方法も提供する必要があります。

5.1.1 SW1 割込みを使用するシングルスレッドアプリケーション

CYW943907AEVAL1F ボードの「SW1」とマークされたボタンを使用して、APP CPU をスリープまたはディープスリープからウェイクアップする割込みをトリガーするには、次の手順を実行してください。

1. SW1 イベントを有効にして、押されたときに割込みを生成してください。

Code Listing 4 スリープ/ディープスリープからウェイクアップする割込みのためにボタン押下を有効にする

```
#include "cyhal_gpio.h"

    /* Configure GPIO button for an interrupt to wake from sleep / deepsleep. */
    cyhal_gpio_init(CYBSP_SW1, CYHAL_GPIO_DIR_INPUT, CYHAL_GPIO_DRIVE_PULLUP,
                    CYBSP_BTN_OFF);
    cyhal_gpio_enable_event(CYBSP_SW1, CYHAL_GPIO_IRQ_FALL, 7, true);
```

2. スリープ (またはディープスリープ) 関数を呼び出してください。

Code Listing 5 スリープ関数の呼び出し

```
#include "cyhal_syspm.h"

    /* Call sleep – must wake through interrupt */
    cyhal_syspm_sleep();
```

Code Listing 6 ディープスリープ関数の呼び出し

```
#include "cyhal_syspm.h"

    /* Call deepsleep – must wake through interrupt */
    cyhal_syspm_deepsleep();
```

3. ボードの SW1 を押して、スリープまたはディープスリープを終了してください。

5.1.2 ティックレススリープまたはディープスリープを使用するシングルスレッドアプリケーション

次のコードリストに、低電力タイマー (lptimer) とティックレススリープ関数を使用して、APP CPU が指定された時間だけスリープまたはディープスリープモードに入ることができるようにする手順を示します。APP CPU スリープモードは、割込みにより指定された時間より早く終了する場合があります。actual_sleep_ms 引数で返される値は、APP CPU がスリープモードまたはディープスリープモードであった時間 (ミリ秒単位) を表します。

1. ティックレススリープまたはディープスリープを使用するために、低電力タイマーを初期化してください。

Code Listing 7 低電力タイマーの初期化

```
#include "cyhal_lptimer.h"
cyhal_lptimer_t lptimer;

    /* Initialize the low power timer */
    cyhal_lptimer_init(&lptimer);
```

2. APP CPU をスリープまたはディープスリープにしてください。

Code Listing 8 ティックレススリープの使用

```
uint32_t desired_ms;
uint32_t actual_sleep_ms;

    /* Call tickless sleep – will wake after specified time */
    desired_ms = 1000;
     cyhal_syspm_tickless_sleep(&lptimer, desired_ms, &actual_sleep_ms);
```

Code Listing 9 ティックレスディープスリープの使用

```
uint32_t desired_ms;
uint32_t actual_sleep_ms;

    /* Call tickless deepsleep – will wake after specified time */
     desired_ms = 1000;
     cyhal_syspm_tickless_deepsleep(&lptimer, desired_ms, &actual_sleep_ms);
```

デバイスは、指定された desired_ms 時間後にウェイクアップします。

5.2 マルチスレッドアプリケーション

マルチスレッドアプリケーションでは、ユーザーアプリケーションはスリープ関数を直接呼び出しません。これらは、アイドルタスクで RTOS から呼び出されます。Code Listing 10 と Code Listing 11 に、RTOS がスリープモードとディープスリープモードをイネーブルする方法を示します。

Code Listing 10 APP CPU アイドル時に RTOS のスリープを有効にする

```
<root of application>/bsps/TARGET_APP_CYW954907AEVAL1F/cybsp.h

#ifndef CY_CFG_PWR_SYS_IDLE_MODE
#define CY_CFG_PWR_SYS_IDLE_MODE        (CY_CFG_PWR_MODE_SLEEP)
#endif
```

Code Listing 11 APP CPU アイドル時に RTOS のディープスリープ呼び出しを有効にする

```
<root of application>/bsps/TARGET_APP_CYW954907AEVAL1F/cybsp.h

#ifndef CY_CFG_PWR_SYS_IDLE_MODE
#define CY_CFG_PWR_SYS_IDLE_MODE        (CY_CFG_PWR_MODE_DEEPSLEEP)
#endif

#define CY_CFG_PWR_DEEPSLEEP_LATENCY    (24UL)  /* Min. 24 for this device */
```

5.3 ディープスリープのロック/ロック解除

ユーザーアプリケーションでスレッドを遅延させる必要があるが、デバイスをディープスリープにしたくない場合は、保護されたコードの周辺でロックとロック解除を呼び出してください。

ディープスリープをロックまたはロック解除するには、次の関数を使用してください。

- cyhal_syspm_lock_deepsleep(): この関数は、ロック解除関数が呼び出されるまで、デバイスがディープスリープに入ることを禁止します。この関数は、複数のスレッドがこの関数を呼び出せるように、カウンタを使用します。すべてのスレッドがディープスリープのロックを解除するまで、デバイスはディープスリープになりません。
- cyhal_syspm_unlock_deepsleep(): この関数はカウンタをデクリメントします。すべてのスレッドがロック解除を呼び出すと、RTOS は、正しい条件が満たされたときに、デバイスをディープスリープにし始めることができます。

Code Listing 12 ディープスリープのロック

```
/* Lock device from entering deepsleep for this section of code */
cyhal_syspm_lock_deepsleep();

    /*
     * Deepsleep will not occur in this section of code
     */

/* Allow deepsleep again */
cyhal_syspm_unlock_deepsleep();
```

参考資料

[1] Hardware Abstraction Layer Power Management API

[2] インフィニオン アプリケーションノート AN214828 Power Consumption Measurements

[3] Low Power Current Measurements Using CYW943907AEVAL1F

改訂履歴

版数	発行日	変更内容
**	2024-11-19	本版は英語版 002-37426 Rev.** について、インフィニオン DEVELOPER COMMUNITY の参画者によって日本語に翻訳されたドキュメントです。

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2024-11-19
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2024 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
002-40871 Rev. **

重要事項
本手引書に記載された本製品の使用に関する手引きとして提供されるものであり、いかなる場合も、本製品における特定の機能性能や品質について保証するものではありません。本製品の使用の前に、当該手引書の受領者は実際の使用環境の下であらゆる本製品の機能及びその他本手引書に記された一切の技術的情報について確認する義務が有ります。インフィニオンテクノロジーズはここに当該手引書内で記される情報につき、第三者の知的所有権の不侵害の保証を含むがこれに限らず、あらゆる種類の一切の保証および責任を否定いたします。

本文書に含まれるデータは、技術的訓練を受けた従業員のみを対象としています。本製品の対象用途への適合性、およびこれら用途に関連して本文書に記載された製品情報の完全性についての評価は、お客様の技術部門の責任にて実施してください。

警告事項
技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。
インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないこと予めご了承ください。